

TDB-ACC-NO: NN9408381

DISCLOSURE TITLE: Method of Sorting Dates and Time Allowing for Wrapping

PUBLICATION-DATA: IBM Technical Disclosure Bulletin, August 1994, US

VOLUME NUMBER: 37

ISSUE NUMBER: 8

PAGE NUMBER: 381 - 382

PUBLICATION-DATE: August 1, 1994 (19940801)

CROSS REFERENCE: 0018-8689-37-8-381

DISCLOSURE TEXT:

Disclosed is a method of sorting dates, times, or other data which may wrap the counter. The method does not require specification of an arbitrary window which is assumed to contain the

times, but rather uses the data and the information that the numbers

are all given modulo something in order to determine the correct ordering. Specific applications include the problems of sorting two-digit years which may cross a century boundary, months which may

encompass the end of the year, and times which encompass midnight.

The algorithm applies to sorting all cyclical groups of numbers; the

sorted numbers need not represent time.

- The problem solved may be illustrated with dates in the neighborhood of the year 2000. The years labelled '03, '05, '02, '99, '96, would properly be sorted chronologically as '96, '99, '02,

'03, '05. A conventional sorting routine would sort them as '02, '03, '05, '96, '99. To obtain the correct sorting, a traditional approach is to pick an arbitrary year and assume that all dates are

after that year, pre-append the appropriate century, and then use a

conventional sorting algorithm. For example, one might assume that

all the specified dates are after the year 1975. So '96 must mean

1996 and '03 must mean 2003. The disclosed method avoids this

arbitrary designation of a date. Instead, it examines the data, taking into account the possibility of wrapping, finds the largest gap, and then shifts the sort so that the number after the largest gap comes first.

For the dates '02, '03, '05, '96, '99, the successive differences are 1, 2, 91, 3, and 3 years. The second difference of 3 is found by using the understanding that the original numbers are given modulo 100, i.e. using circular subtraction. Since the third difference, 91, is the largest, the fourth date, '96, should come first, and the correct sorting is then '96, '99, '02, '03, '05.

- Let MAXNUM be the largest possible number representable, that is, one less than the modulo base. For example, the largest year representable with two digits is '99. We wish to sort the array A(i), taking into account the fact that the numbers A(i) may wrap.

The method is

1. Sort the numbers A(i) using a conventional sorting routine.
2. Find the largest gap in the sorted numbers, using circular subtraction. For N numbers, this is illustrated with the pseudocode

```
DIFFMAX = A(1) + MAXNUM + 1 - A(N)
```

```
STARTI = 1
```

```
do I=1 to N-1
```

```
    DIFF = A(I+1) - A(I)
```

```
    if DIFF > DIFFMAX then do
```

```
        STARTI = I + 1
```

```
        DIFFMAX = DIFF
```

```
    end
```

```
end
```

The largest gap, of size DIFFMAX, occurs before A(STARTI).

In

case of ties, this algorithm chooses the first occurrence of

the

largest gap.

- 3.

Shift the sorted numbers so that the number after the largest gap

comes first. That is

```
J = STARTI
```

```
do I=1 to N
```

```
    B(I) = A(J)
```

```
    J = J + 1
```

```
    if J > N then J = 1
```

```
end
```

The array B(i) now contains the numbers, sorted by what is likely to be the correct order when the possible wrapping is taken into account. The array B(i) is introduced only for illustrative purposes. Other means may be used to shift the array so that the first element becomes what was A(STARTI) and the circular order is maintained.

- This algorithm will produce proper orderings as long as the numbers are reasonably close together in the circular sense. It has the advantage that the algorithm itself will never become out of date. Whether it is appropriate to use depends on the anticipated scatter of the circular data. Years do not typically have a great deal of scatter, which is why notation which drops the century has come to be employed. The algorithm may be expected to work well with such data. The key requirement is that the numbers fall into some reasonable span. A person contemplating sorting the dates '05, '95, '99, and '01 would probably assume that '99 comes before '01.

How would they know? They would know of the possibility of wrapping, and guess that wrapping did occur for the little numbers because with that assumption the total span of years is much smaller. The above algorithm accomplishes this minimizing of the total span of years.

SECURITY: Use, copying and distribution of this data is subject to the restrictions in the Agreement For IBM TDB Database and Related Computer Databases. Unpublished - all rights reserved under the Copyright Laws of the United States. Contains confidential commercial information of IBM exempt from FOIA disclosure per 5 U.S.C. 552(b)(4) and protected under the

Trade  
Secrets Act, 18 U.S.C. 1905.

COPYRIGHT STATEMENT: The text of this article is Copyrighted (c) IBM Corporation 1994. All rights reserved.